Polynomial ideal theoretic methods in discrete event, and hybrid dynamical systems

Michel Le Borgne, Albert Benveniste, Paul Le Guernic

UNITÉ DE RECHERCHE
INRIA-RENNES

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P. 105
78153 Le Chesnay Cedex
France
Tél. (1) 39 63 55 11

Rapports de Recherche

N° 1091

Programme 1
Programmation, Calcul Symbolique
et Intelligence Artificielle

POLYNOMIAL IDEAL THEORETIC METHODS IN DISCRETE EVENT, AND HYBRID DYNAMICAL SYSTEMS

Michel Le BORGNE
Albert BENVENISTE
Paul Le GUERNIC

Septembre 1989

RR.1091

IRISA

INSTITUT DE RECHERCHE EN INFORMATIQUE
ET SYSTÈMES ALÉATOIRES

Campus Universitaire de Beaulieu
35042 - RENNES CÉDEX
FRANCE
Téléphone : 99 36 20 00
Télex : UNIRISA 950 473 F
Télécopie : 99 38 38 32

Publication Interne n° 483 - Juillet 1989 - 20 Pages

Polynomial Ideal Theoretic methods in Discrete Event, and Hybrid Dynamical Systems.

Michel Le Borgne, Albert Benveniste, Paul Le Guernic
IRISA, Campus de Beaulieu
35042 RENNES CEDEX

Abstract

This short paper is a first attempt to rephrase selected properties relevant to Discrete Event or Hybrid Dynamical Systems in terms of nonlinear systems over finite fields. Such systems play a central role in the compilation of the SIGNAL language.

Méthodes de géométrie algébrique effective pour les systèmes dynamiques hybrides ou à évènements discrets.

Résumé

On présente ici une transcription de l'étude des systèmes à évènements discrets ou hybrides en termes de calculs d'idéaux de polynômes à plusieurs indéterminées sur des corps finis. De tels systèmes jouent un rôle fondamental dans la compilation du langage synchrone SIGNAL.

CENTRE NATIONAL DE LA RECHERCHE SCIENTIFIQUE
(L. A. 227)
UNIVERSITÉ DE RENNES 1 I. N. S. A. DE RENNES

INSTITUT NATIONAL DE RECHERCHE
EN INFORMATIQUE ET EN AUTOMATIQUE
(LABORATOIRE DE RENNES)

1 Introduction

Discrete Event Dynamical Systems (DEDS) have been introduced as a theoretical framework for the study of flexible manufacturing and related systems by Wonham and Ramadge [1] [2], and have been widely studied since their introduction. Roughly speaking, DEDS are finite state transition systems which are observed and can be controlled by the language generated by the labels that are attached to each transition, regardless of the precise meaning of these labels. On the other hand, *Hybrid Dynamical Systems (HDS)* theory has been introduced in [3] and [4] to handle synchronization, logic, and their interconnections to numerics in dynamical systems. HDS theory covers areas such as real-time process control, real-time signal processing systems, and more generally, C^3-systems.

In this paper, we want 1/ to establish a connection between these two theories and the study of non linear dynamical systems over finite fields, and 2/ to show how polynomial ideal theory techniques can be used to solve various questions that are fundamental in DEDS supervision in the sense of Ramadge and Wonham, and in the compilation of the SIGNAL language.

This paper is a shortened version of a full paper in preparation.

2 Relating DEDS, HDS, and dynamical systems over finite fields.

2.1 A brief presentation of the SIGNAL language.

For a complete presentation and motivation of the language, we refer the reader to [3] and [4]. SIGNAL is a language to specify, analyse, and execute Hybrid Dynamical Systems, i.e. general dynamical systems of the form

$$\begin{aligned} \xi_{t+1} &= f(\xi_t, y_t) \\ 0 &= g(\xi_t, y_t) \end{aligned}$$

where the variables ξ_t and y_t are both vector valued, and $t = 1, 2, 3, \ldots$. The functions f and g are completely general. ξ_t is the state vector, some of the components of y_t may be viewed as inputs while the others are considered as outputs, so that the above dynamical system is *implicit* or *relational*. Now, each variable, in addition to the normal value it takes in its range, can also take a special value denoted by $\perp$, representing the *absence* of data at that instance (which means that nothing is visible for an observer of this single signal only). Therefore, an infinite time sequence of an integer variable (we shall refer to as an integer *signal* in the sequel) may look like $1, -4, \perp, \perp, 4, 3, \ldots$. It is shown in [3] and [4] that HDS can be efficiently specified using the SIGNAL language, which is composed of the following five instructions:

```
p(b1,...,bn), y := f(x1,...,xn)
y := x $ init y0
y := x when b
y := u default v
P|Q
```

We shall now comment these instructions, and introduce at the same time their encoding using dynamical systems over the finite field $\mathcal{F}_3$ of integers modulo 3.

Instructions `a and b =` *true* `, y := u+v`
The first one is a particular case of the generic instruction `p(b1,...,bn)` where `p(...)` is any relation on booleans. It extends to signals the mentioned relation. First, the signals `a, b` must *have the same clock* (i.e. they must be based on the same time index), and second, when they are present, their actual values must satisfy the mentioned relation on boolean values.

The second instruction is a particular case of the generic form `y := f(x1,..,xn)`. It extends to signals the mentioned function on data. First, all signals `y, u, v` must have the same clock, and second, when they are present, the value carried by `y` must be equal to `u+v`.

The reasoning mechanisms of SIGNAL can handle (i) the presence/absence, (ii) the boolean values since they are important in modifying clocks, and (iii) *dependence graphs* to encode data dependencies in non boolean functions in order to avoid to solve general implicit systems of equations. Hence three labels are needed to encode *absent*, *true*, *false*: the finite field $\mathcal{F}_3$ of integers modulo 3 is used for this purpose:

$$absent \rightarrow 0,\ true \rightarrow +1,\ false \rightarrow -1$$

Using this mapping, the two instructions `a and b = ` *true* ` , y := u+v` are respectively encoded as follows:

$$\begin{aligned} a^2 &= b^2,\ ab + a + b = 0 & (1) \\ y^2 &= u^2 = v^2,\ \mathtt{u} \xrightarrow{y^2} \mathtt{y} \xleftarrow{y^2} \mathtt{v} & (2) \end{aligned}$$

In these equations, the variables $a, x, \ldots$ refer to infinite sequences of data in $\mathcal{F}_3$ with time index implicit. The first equation of (1) expresses that the two signals `a` and `b` must have the same clock, while the second one encodes the particular boolean relation (the first equation is here a consequence of the second one). The first equation of (2) again expresses that all signals must have the same clock, while the labelled graph expresses that the mentioned data dependencies hold when $y^2 = 1$, i.e. when all signals are present (this is referred to as the *conditional dependence graph* since signals may be related via different dependencies at different clocks).

Since our purpose here is not to fully investigate the interconnection of clocks and dependencies (see [3] and [4] for an extensive discussion on this topic), we shall concentrate in the sequel on *boolean* instructions.

Instruction `y := x $ init y0`
This instruction simply means $\forall t : y_t = x_{t-1}, y_0 = \mathtt{y0}$ where the time index t enumerates the instants where both signals `x, y` are present: this is just a shift register. Boolean shift registers are encoded as follows

$$\begin{aligned} \xi' &= (1 - x^2)\xi,\ \xi_0 = \mathtt{yo} & \\ y &= x^2 \xi & (3) \end{aligned}$$

where the variable ξ is the state, and ξ' denotes its next value according to any (hidden) time index which is more frequent than the index t above. This is a nonlinear dynamical system. Notice that receiving $x = 0$ (i.e. the input being absent at the considered instant) does not change the state nor produces any output (since $y = 0$ holds also in this case).

Instruction `y := x when b.`
The value of `y` equals the value of `x` when `x` and `b` are present, and `b`= *true*. This gives:

$$y = x(-b - b^2) \tag{4}$$

Instruction `y := u default v.`
The value of `y` equals the value of `u` when `u` is present, and by default the value of `v` when `v` is present. Otherwise nothing is delivered. This gives the coding:

$$y = u + v(1 - u^2) \tag{5}$$

Instruction `P|Q`
Here, `P`, `Q` denote SIGNAL *programs*, i.e. already defined sets of instructions or programs, and `P|Q` denotes the new program obtained by combining the already defined ones by considering that signals with identical names are identical (just as in mathematics). The coding of this instruction is obtained by considering together as a single dynamical system the two dynamical systems associated to `P` and `Q`.

The general form. By combining together the particular dynamical systems of eqns (1) (2) (3) (4) (5), we obtain the general form of dynamical system over $\mathcal{F}_3$

$$\begin{aligned} x_{t+1} &= P(x_t, y_t) \\ 0 &= Q(x_t, y_t) \\ 0 &= Q_0(x_0) \end{aligned} \tag{6}$$

Notations: vectors and polynomial functions. In eqn. (6), x, y are vectors with components in $\mathcal{F}_3$, the components of x will be denoted by $x(i)$ and those of y by $y(j)$. P, Q denote polynomial vector functions in the components of x and y.

The components of x encode the state of the boolean memories, while the components of y encode the signals involved in the program. The time index t is any time index which is more frequent than the clock of all the signals involved in the program. Finally, the last equation specifies the initial condition; this last equation will be omitted for the sake of simplicity in the sequel. Sequences (x_t, y_t) satisfying the constraint (6) will be termed *admissible*; selected components of an admissible sequence will also be termed admissible. It is easily seen from eqn (3) that dynamical systems arising from SIGNAL programs satisfy the following additional property:

$$y_t = 0 \Longrightarrow \begin{cases} x_{t+1} = x_t \ : \ \text{no change in the state} \\ 0 = Q(x_t, y_t) \text{ holds} \end{cases} \tag{7}$$

This expresses that it is always allowed for a SIGNAL program to do *nothing*: no input received and no output delivered (encoded by $y_t = 0$), no change in the state (encoded by $x_{t+1} = x_t$).

2.2 Relating DEDS and dynamical systems over finite fields.

Several DEDS frameworks have been proposed. Here we choosed to analyse [5] (where Büchi automata are considered) and we refer the reader to this paper for the notations and definitions. Referring to section II.C of this paper, we consider a *generator*, consisting of a state set $\mathcal{Q}$, an initial state q_0, and a transition function $\delta : \Sigma \times \mathcal{Q} \longrightarrow \mathcal{Q}$ (in general, a partial function). Σ is the set of *events*, and consists of labels. Then, *controlled* DEDS are defined next, following section II.D of [5]. Partition $\Sigma = \Sigma_u \cup \Sigma_c$ into uncontrollable and controllable events. An "admissible" (in the sense of this reference) control law consists of a subset $\gamma \subseteq \Sigma$ satisfying $\Sigma_u \subseteq \gamma$, such a control law specifies the events that are *enabled*. Controlled DEDS can be encoded in our framework as follows.

Set $\Xi = \mathcal{F}_3^{\mathcal{Q}}$ and consider the map $\mathcal{Q} \rightarrow \Xi$ which assigns to $q \in \mathcal{Q}$ the point of Ξ with q-th coordinate equal to $+1$ and all other equal to -1. Similarly, write $\mathcal{Y} = \mathcal{F}_3^{\Sigma}$, and assign to $\sigma \in \Sigma$ the point in $\mathcal{Y}$ with σ-th coordinate $= 1$ and the other $= 0$. Then, it is easily verified that any controlled DEDS can be encoded as a dynamical system of the form (6).

3 Some basic problems and their geometric translation.

3.1 Testing for inputs, and observability.

As SIGNAL is a *declarative* language, the notion of input/output may be ignored. Each SIGNAL instruction defines a relation between signals which is translated into an equation between clocks in the dynamical system over $\mathcal{F}_3$ corresponding to the program. If one wants some signals to be considered as inputs and others as outputs, there must be a function mapping input clocks into output clocks. This can be seen as an observability problem since one wants to know whether *the input clocks completely determine the output clocks.* Consider the dynamical system

$$\begin{aligned} x_{t+1} &= P(x_t, y_{1,t}, y_{2,t}) \\ 0 &= Q(x_t, y_{1,t}, y_{2,t}) \end{aligned} \tag{8}$$

It is desired that the second equation of (8) could be rewritten as follows:

$$y_{2,t} = \tilde{Q}(x_t, y_{1,t}) \tag{9}$$

3.2 Synthesis: the problem of exact model following control.

The classical problem of exact model following control is: *given the dynamical system* (8), *and a second one considered as the reference model,*

$$\begin{aligned} x_{m,t+1} &= P_m(x_{m,t}, y_{1,t}) \\ 0 &= Q_m(x_{m,t}, y_{1,t}) \end{aligned} \tag{10}$$

is it possible to find a relation $K(x_m, x, y_1, y_2) = 0$ such that all admissible sequences $(y_{1,t})$ of the system obtained by combining (8) and this relation, are admissible sequences of the reference model (10). If this additional relation is of the form $y_2 = \tilde{K}(x_m, x, y_1)$, the exact model following control is just performed via a *feedback* in the usual sense.

3.3 A proof system: inspecting a system via another one.

Given the system (8), we wish to "inspect" a subset of the signals involved in the program, for instance the $(y_{1,t})$. What we need for this purpose is a "compact" description of the behaviour of these components: this will be obtained by *building a dynamical system*

$$\begin{aligned} x_{c,t+1} &= P_c(x_{c,t}, y_{1,t}) \\ 0 &= Q_c(x_{c,t}, y_{1,t}) \end{aligned} \tag{11}$$

possessing as only admissible sequences the sequences $(y_{1,t})$ *that are admissible for (8).*

A further generalization can be obtained as follows. Assume we are interested in keeping track of the components $y_{1,t}$ only when they satisfy a particular (dynamical) relation, specified (for instance) by a system of the form (10). Then we only have to answer the same question as above for the combination of the two systems (8) and (10).

4 A sample of properties of DEDS and their effective checking.

4.1 A toolbox in polynomial ideal theory.

4.1.1 Algebraic varieties and polynomial ideals.

The dynamical systems (6) involve two kind of equations: an evolution equation $x_{t+1} = P(x_t, y_t)$ and a set of constraints $Q(x_t, y_t) = 0$. These constraints define an algebraic variety in $\mathcal{F}_3^{dim(x)} \times \mathcal{F}_3^{dim(y)}$. It is well known that it is possible to replace the set of equations $Q(x, y) = 0$ by the set of all linear combinations, with polynomial coefficients, of the equations without changing the algebraic variety. This new set of equations is the *ideal* generated by the polynomials defining the components of Q. We need to refresh our notations to handle polynomials.

Notations: rings of polynomials, ideals. The notation X will denote a collection of formal variables $X(i)$, and similarly for Y, etc... Referring to (6), the notation $Q(X,Y)$ will denote a *collection* of polynomials in the variables $X(i), Y(j)$. The ring of the polynomials in these variables with coefficients in $\mathcal{F}_3$ will be denoted by $\mathcal{F}_3[X,Y]$, this notation extends to any set $Z, T, ...$ of formal variables. Let us denote by $< Q(X,Y) >$ the ideal spanned by the mentionned collection of polynomials, and by $\mathcal{V}(Q(X,Y))$ the algebraic variety associated with the set of equations $Q(x,y) = 0$. We have:

$$\mathcal{V}(Q(X,Y)) = \mathcal{V}(< Q(X,Y) >) \tag{12}$$

On the other side, given a subset S of $\mathcal{F}_3^{dim(x)} \times \mathcal{F}_3^{dim(y)}$, the set of polynomials R in $\mathcal{F}_3[X,Y]$ such that $R(s) = 0$ for all $s \in S$, is an ideal of $\mathcal{F}_3[X,Y]$ denoted $\mathcal{I}(S)$.

Now, how is an ideal $\underline{a}$ related to $\mathcal{I}(\mathcal{V}(\underline{a}))$? The best we can hope is $\mathcal{I}(\mathcal{V}(\underline{a})) = \underline{a}$ for all ideal $\underline{a}$. Unfortunately this is not true in general, but thanks to properties of the field $\mathcal{F}_3$, we have:

$$\mathcal{I}(\mathcal{V}(\underline{a})) =< \underline{a}, X^3 - X, Y^3 - Y > \tag{13}$$

where $X^3 - X$ (resp. $Y^3 - Y$) stands for the set of polynomials $X^3(i) - X(i)$ (resp. $Y^3(i) - Y(i)$). This result means that we have only to add to every ideal the set of everywhere null polynomials to obtain a perfect correspondence between ideals and algebraic varieties. So, from now on, $< Q(X,Y) >$ will denote $< Q(X,Y), X^3 - X, Y^3 - Y >$ and *all the ideals we shall consider in the sequel will be augmented with the associated polynomials* $X^3 - X$*, etc....*

On the other hand, for all set S we have

$$\mathcal{V}(\mathcal{I}(S)) = S$$

A nice consequence of the former results is that any ideal associated to varieties may be considered as an ideal of the quotient ring

$$\mathcal{F}_3[X,Y] \;/ < X^3 - X, Y^3 - Y >$$

Since this ring is a finite dimensional vector space over $\mathcal{F}_3$ with dimension $3^{(dim(x)+dim(y))}$, *strictly increasing chains of ideals are finite with length bounded by* $3^{(dim(x)+dim(y))}$.

4.1.2 Projections and morphisms.

Consider multivariables Z and T. A morphism $\theta : \mathcal{F}_3^{dimZ} \rightarrow \mathcal{F}_3^{dimT}$ is a mapping such that each component is a polynomial function. The evolution equation of a dynamical system over $\mathcal{F}_3$ is a morphism. In fact, due to the finiteness of the involved spaces, every mapping $\mathcal{F}_3^{dimZ} \rightarrow \mathcal{F}_3^{dimT}$ is a morphism. This can be easily proved using the Lagrange interpolation polynomials.

Given a morphism θ as above with components $(\theta_1(Z), \dots, \theta_{dimT}(Z))$, we define a ring homomorphism $\theta^* : \mathcal{F}_3[T] \rightarrow \mathcal{F}_3[Z]$ by:

$$p \in \mathcal{F}_3[T] : \quad \theta^*(p(T_1, \dots, T_{dimT})) = p(\theta_1(Z), \dots, \theta_{dimT}(Z)) \tag{14}$$

θ^* is known as the *comorphism associated to* θ. The proofs of the following relations are straighforward:

$$\begin{aligned} \mathcal{I}(\theta(\mathcal{V}(\underline{a}))) &= \theta^{*-1}(\underline{a}) \\ \mathcal{V}(\theta^*(\underline{b})) &= \theta^{-1}(\mathcal{V}(\underline{b})) \end{aligned} \tag{15}$$

for all ideals $\underline{a}$ in $\mathcal{F}_3[Z]$ and all ideals $\underline{b}$ in $\mathcal{F}_3[T]$.

The projections onto the components of the spaces $\mathcal{F}_3^n$ are morphisms of particular interest when observability is analysed. In particular we have the notion of *conditional independence* of two subsets of components for an algebraic variety:

Definition 1 *Given an algebraic variety V included in the space $\mathcal{F}_3^{dim(X)} \times \mathcal{F}_3^{dim(Y)} \times \mathcal{F}_3^{dim(Z)}$, we will say that Y and Z are conditionally independent given X, if for all x, y, z:*

$$(x, y) \in proj_{X,Y}(V) \;\&\; (x, z) \in proj_{X,Z}(V) \;\Rightarrow (x, y, z) \in V$$

where $proj_X$ denotes the projection onto the components corresponding to X.

This property is equivalent to $V = (proj_{X,Y} \times \mathcal{F}_3^{dim(Z)}) \cap (proj_{X,Z} \times \mathcal{F}_3^{dim(Y)})$ or translated into ideal properties as follows:

$$\mathcal{I}(V) = \mathcal{I}(V) \cap \mathcal{F}_3[X, Y] \;+\; \mathcal{I}(V) \cap \mathcal{F}_3[X, Z]$$

4.1.3 A quick review of Gröbner bases as an effective tool.

If we want to use elementary algebraic geometry to study DEDS or HDS we need an effective tool to solve problems such as computing the ideal corresponding to the projection of a variety on some components, or checking whether a polynomial belongs to a given ideal. An ideal can be handled only via a set of generators. So we ask for "canonical" sets of generators which reflect faithfully the properties of ideals.

In polynomial rings in *one variable* over a field, each ideal is generated by a single polynomial. The Euclidean division by this generator is the basic tool for solving various ideal problems. The basic step in the Euclidean division consists in replacing the dividend by a new one of lower degree: this is done by substracting a suitable multiple of the divisor. This procedure is applied as long as possible, and the ordering of the monomials according to increasing degrees plays an essential role.

In order to extend this operation to polynomials with several variables, we need a total order defined on monomials which is compatible with the product. That is to say, for all monomials m_1, m_2, m:

$$m_1 \prec m_2 \Rightarrow mm_1 \prec mm_2 \tag{16}$$

Orders with this property are for instance the lexicographical order shown here for 2 variables X, Y:

$$\begin{aligned} 1 \prec X \prec X^2 \prec X^3 \prec \ldots \prec Y \prec XY \prec X^2Y \prec \ldots \\ \ldots \prec Y^2 \prec XY^2 \prec X^2Y^2 \prec \ldots \end{aligned} \tag{17}$$

or the total degree ordering:

$$1 \prec X \prec Y \prec X^2 \prec XY \prec Y^2 \prec X^3 \prec X^2Y \prec \ldots \tag{18}$$

If q is a polynomial, let us denote $lm(q)$ the greatest monomial in q with respect to some compatible order and $lc(q)$ the coefficient of $lm(q)$. If $\mathcal{Q}$ is a finite family of polynomials and p a polynomial, p can be *reduced* modulo $\mathcal{Q}$ if there exists a polynomial q in $\mathcal{Q}$ such that $lm(q)$ divides $lm(p)$. This implies $lm(q) \prec lm(p)$ using the compatibility of the order with the product. Then p reduces to p_1 modulo $\mathcal{Q}$ if:

$$p_1 = p - lc(p)\frac{lm(p)}{lm(q)}q \tag{19}$$

This reduction procedure may be carried on, but eventually stops. Notice that the reduction does not change the class of p modulo the ideal generated by Q and the reduction procedure gives a polynomial which is equivalent to the initial one (modulo Q) and minimal with respect to the considered order.

The question is now: is this reduced polynomial unique? The answer is *not* in general. The essential reason being that, at each step of the reduction procedure, a choice must be made among the polynomials of Q which can be used for the reduction. There is no evidence that the final reduced polynomial is independent of these choices and in general *it is not.*

Here come Gröbner bases for rescue!... It turns out that for every ideal $\underline{a}$ and every compatible order $\prec$ on monomials, it is possible to find a set of generators $G(\underline{a})$ such that *every reduction performed modulo $G(\underline{a})$ is independent of the choices made at each step.* For more details and the construction of Gröbner bases, the reader is referred to [6]. Gröbner bases are not unique, but so-called *"reduced"* Gröbner bases are unique, and will be implicitely used in the next criteria.

Among several problems in ideal theory solved by Gröbner bases, the following are useful for our purpose:

- $p \in \underline{a}$ if and only if p reduces to 0 modulo any Gröbner basis $G(\underline{a})$.
- If $\underline{a}$ is an ideal in $\mathcal{F}_3[X, Y]$ then $G(\underline{a} \cap \mathcal{F}_3[X]) = G(\underline{a}) \cap \mathcal{F}_3[X]$ for any Gröbner basis associated with a lexicographical order such that $X \prec Y$ (this notation stands for short to mention that $X(i) \prec Y(j) \forall (i, j)$).
- The following property does not hold in general, but is valid for the particular ideals (with $X^3 - X$ polynomials) we consider here. With the notations of definition 1, Y and Z are conditionally independent given X if and only if Y and Z do not appear simultaneously as variables in any polynomial of any Gröbner basis for any lexicographical order such that $X \prec Y \prec Z$ or $X \prec Z \prec Y$.

The two first properties are well known in Gröbner bases theory. The third property is less classical and its proof is deferred to a forthcoming paper.

4.2 Studies on observability.

4.2.1 Static observability of systems and application to the first basic problem.

In order to introduce the notions of input and output let us split the set of variables y into two sets y_1 and y_2. The dynamical system is then:

$$\begin{aligned} x_{t+1} &= P(x_t, y_{1,t}, y_{2,t}) \\ 0 &= Q(x_t, y_{1,t}, y_{2,t}) \end{aligned} \tag{20}$$

We formalize the notion of input-output as follow:

Definition 2 *The pair (Y_1, Y_2) is an input-output pair for the dynamical system (20) if there exists a mapping f from $\mathcal{F}_3^{dim x} \times \mathcal{F}_3^{dim y_1}$ to $\mathcal{F}_3^{dim y_2}$ such that, for all (x, y_1) in $proj_{X,Y_1}(\mathcal{V}(Q))$, $y_2 = f(x, y_1)$ is the unique solution of the equation $Q(x, y_1, y_2) = 0$.*

The function f is necessarily a morphism (Cf. (4.1.2)).

Theorem 1 *The following assertions are equivalent:*

1. *The pair (Y_1, Y_2) is an input-output pair for the dynamical system (20).*
2. *There exists a set of generators of the ideal $< Q >$ composed of polynomials in $\mathcal{F}_3[X, Y_1]$ and of polynomials of the form $Y_2 - R(X, Y_1)$.*
3. *The reduced Gröbner basis of $< Q >$ for any lexicographic order satisfying $\{X, Y_1\} \prec Y_2$ possesses the same property as the above set of generators.*

Sketch of the proof: $3 \Rightarrow\ 2 \Rightarrow\ 1$ is obvious. $1 \Rightarrow\ 2$ follows from the fact that every map is polynomial. $2 \Rightarrow 3$ is a consequence of the Buchberger algorithm for building Gröbner bases (The proof is deferred to a forthcoming full paper).

Remark: The concept of input/output pair can be generalized to the notion of functional dependency. With the above notations, Y_2 is termed *functionally dependent* on Y_1, if there exists a function f such that any solution (x, y_1, y_2) of $Q(X, Y_1, Y_2) = 0$ must satisfy $y_2 = f(y_1)$. A theorem similar to theorem 1 can be stated, where the functional dependency implies the presence of polynomials $Y_2 - R(Y_1)$ in some generator sets of $< Q >$.

4.2.2 Liveness and deadlocks, and solution of the second basic problem.

Considering a dynamical system (6), liveness is naturally defined as follows:

Definition 3 *A state vector x is alive if there exists a vector y such that $(x, y) \in \mathcal{V}(Q)$. A system is termed alive if for each state x and each y such that $(x, y) \in \mathcal{V}(Q)$, $P(x, y)$ is a live state.*

In other words, a state is termed alive if it belongs to $proj_X(\mathcal{V}(Q))$. A system is alive iff $P(\mathcal{V}(Q)) \subseteq proj_X(\mathcal{V}(Q))$. Translating into corresponding ideals properties with the help of relations (15), we get:

Proposition 1 *A dynamical system is alive if an only if*

$$P^*(< Q >) \quad \subseteq \quad < Q > \cap \mathcal{F}_3[X]$$

where P^* is the comorphism associated with P.

A SIGNAL dynamical system is always alive as for every state x, $(x, 0) \in \mathcal{V}(Q)$ (Cf eqn. (7)). States x for which the only autorized value of y is 0 will be called *deadlocked states*. A SIGNAL dynamical system is *deadlock-free* if for each $(x, y) \in \mathcal{V}(Q)$, $P(x, y)$ is not a deadlocked state.

Proposition 2 *A* SIGNAL *dynamical system is deadlock-free if and only if the dynamical system*

$$\begin{aligned} x_{t+1} &= P(x_t, y_t) \\ 0 &= Q(x_t, y_t) \\ 0 &= \prod_{i=1}^{dimY} (y_t^2(i) - 1) \end{aligned}$$

is alive.

The result is obvious since the constraint equation added to the initial SIGNAL system prevents all components of Y from being simultaneously equal to 0.

The liveness concept gives a tool to solve the *exact model following control problem.* Consider the dynamical system obtained by combining (8) and (10). The exact following model control problem is solved if it is possible to add constraints on x, y_1, y_2 such that the resulting dynamical system is alive. Introduce $\underline{c} =< Q(X, Y_1, Y_2), Q_m(X_m, Y_1) >$, and consider the increasing chain of ideals:

$$
\begin{aligned}
c_0 &= \underline{c} \\
c_1 &= < c_0, P^*(c_0) \cap \mathcal{F}_3[X, X_m]) > \\
&\dots \\
c_i &= < c_{i-1}, P^*(c_{i-1} \cap \mathcal{F}_3[X, X_m]) > \qquad (21)
\end{aligned}
$$

This chain must be constant after some $k \geq 0$. It is then straightforward to check that $P^*(c_k) \subseteq c_k \cap \mathcal{F}_3[X, X_m]$. If the largest ideal of the chain is the whole ring, *it is impossible for the dynamical system (8) to follow (10).* In the other case the largest ideal c_k gives the constraints we have to add to (8) in order to force it to behave as wished.

4.2.3 Dynamic observability of systems and application to the third basic problem.

Different kinds of observability concepts can be defined for a dynamical system such as (6). In this section we shall be interested in an approach linked to Büchi automata theory. A dynamical system is a Büchi automaton with $proj_Y(\mathcal{V}(Q))$ as alphabet and the admissible sequences $(y_t)_{t\geq 0}$ as infinite words of the language. Asking whether or not it is possible to reconstruct the state of the dynamical system from the observation of any finite subword of the language refers undoubtedly to an observability concept. Let us call it *dynamic observability by the language.*

In order to study this concept let us introduce the ideals:

$$
\begin{aligned}
\mathcal{OBS}_k = &< Q(X_0, Y_0), Q(X_1, Y_1), \dots, Q(X_k, Y_k), \qquad (22) \\
& X_1 - P(X_0, Y_0), \dots, X_k - P(X_{k-1}, Y_{k-1}) >
\end{aligned}
$$

and:

$$\mathcal{SOBS}_k = \mathcal{OBS}_k \cap \mathcal{F}_3[X_0, Y_0, \ldots, Y_k] \tag{23}$$

for all integer $k \geq 0$.

Theorem 2 *For all dynamical system (6), there exists a smallest integer k_{OBS} such that, for all $k \geq k_{OBS}$, $(X_0, \ldots, X_k)$ and Y_k are conditionnally independent given $(Y_0, \ldots, Y_{k-1})$.*

Sketch of the proof: Given any vector $(y_1, \ldots, y_k)$, consider the fibre over this vector in $\mathcal{V}(\mathcal{SOBS}_k)$ i.e.:

$$\{x_0 / \ (x_0, y_0, \ldots, y_k) \in \mathcal{V}(\mathcal{SOBS}_k)\}$$

The corresponding ideals are obtained by setting $Y_0 = y_0, \ \ldots, Y_k = y_k$ in $\mathcal{SOBS}_k$. It happens that the chain of ideals associated to the fibres over any increasing chain of vectors $(y_0), (y_0, y_1), (y_0, y_1, y_2), \ldots$ is an increasing chain of ideals in $\mathcal{F}_3[X_0]$. Recalling a previous remark, the length of these chains is bounded by some integer $k_{OBS} - 1 \leq 3^{dim(X_0)}$. It is then easily proved that X_0 and $Y_{k_{OBS}}$ are conditionally independent given $Y_0, \ldots, Y_{k_{OBS}-1}$. Carrying out this result to $\mathcal{V}(\mathcal{OBS}_k)$, we get the claimed theorem.

The system is then *observable by the language if and only if $X_{k_{OBS}}$ is functionaly dependent on $(Y_0, \ldots, Y_{k_{OBS}-1})$ in $\mathcal{V}(\mathcal{OBS}_{k_{obs}})$* (cf the remark following the theorem 1). A more important consequence of theorem 2 is the characterisation of the language of any dynamical system. Introduce

$$VL = proj_{(Y_0, \ldots, Y_{k_{OBS}})}(\mathcal{V}(\mathcal{OBS}_{k_{OBS}}))$$

In fact, the ideal associated to this variety is the suitable counterpart in our case of the usual notion of "transfer function". We have:

Theorem 3 *A sequence (y_t) is a word of the language of the dynamical system if an only if every subword $(y_t, \ldots, y_{t+k_{OBS}})$ is in VL.*
Moreover, given an integer k and any variety VL in the space $\mathcal{F}_3^{dim(Y) \times (k+1)}$, the set of sequences $(y_t)_{t \geq 0}$ with the above property is the language of some dynamical system.

Sketch of the proof: The first part of the theorem is a direct consequence of the definition of $\mathcal{OBS}_k$ and theorem 2. For the second assertion introduce $X = (Y_0, \ldots, Y_{k_{OBS}-1})$ and $Y = Y_{k_{OBS}}$. The equation

$$P(x, y) = P((y_0, \ldots, y_{k_{OBS}-1}), y_{k_{OBS}}) = (y_1, \ldots, y_{k_{OBS}})$$

defines the polynomial vector P, and let $Q(X, Y)$ be a set of generators of $\mathcal{I}(\mathcal{V}(L))$. We have defined a dynamical system with the given set of sequences as language: this is a kind of "Auto-Regressive" nonlinear model.

The variety VL associated to the language gives an answer to the problem of inspecting a system by another one. Given the composition of the two systems, compute the reduced Gröbner bases of the ideals $\mathcal{OBS}_k$. Then, based on theorem 2, a direct inspection of these bases provides the index k_{OBS}. Again with the aid of Gröbner bases, compute VL and its projection onto the components corresponding to Y_1, which ultimately yields a complete solution of the inspection problem.

5 Conclusion

This short paper is a first attempt to rephrase selected properties relevant to Discrete Event or Hybrid Dynamical Systems in terms of nonlinear systems over finite fields. The advantage of this approach is twofold: first, it reveals that all operations we have to perform are based on a *single* building block (computing the Gröbner basis of suitable ideals), and second, the framework we obtain is quite familiar to the control community. All this work generalizes immediately to finite fields $\mathcal{F}_p$ for any prime integer p.

References

[1] P.J. Ramadge, W.M. Wonham. *Supervisory Control of a class of Discret Events Processes* , SIAM J. Control and Opt., 25, 1, 1987, 206-230.

[2] P.J. Ramadge, W.M. Wonham. *Controllable Sublanguage of a Given Language*, SIAM J. Control and Opt., 25, 3, 1987, 637-659.

[3] A. Benveniste, P. Le Guernic. *Hybrid Dynamical Systems theory and nonlinear dynamical systems over finite fields,* Proc. of CDC 88, 209-213, Austin 7-9 Dec 1989.

[4] A. Benveniste, B. Le Goff, P. Le Guernic. *Hybrid Dynamical Systems theory and the language SIGNAL.* INRIA research report No 838, 1988.

[5] P.J.G. Ramadge, *Some tractable supervisory control problems for discrete event systems modelled by Büchi automata,* IEEE-AC-34 No 1, 10-20, 1989.

[6] B. Buchberger. *Gröbner Bases: An Algorithmic Method in Polynomial Ideal Theory* N.K. Bose (ed.), Multidimensional Systems Theory, 184-232, D. Reidel Publishing Company.

PI 480 THE MODELLING SYSTEM PYRAMIDE AS AN INTERACTIVE HELP FOR THE GUIDANCE OF THE INSPECTION VEHICLE CENTAURE
Philippe EVEN, Lionel MARCE
22 Pages, Juin 1989.

PI 481 VERS UNE INTERPRETATION QUALITATIVE DE COMPORTEMENTS CINEMATIQUES DANS LA SCENE A PARTIR DU MOUVEMENT APPARENT
Edouard FRANCOIS, Patrick BOUTHEMY
40 Pages, Juin 1989.

PI 482 DEFINITION DE ALPHA : UN LANGAGE POUR LA PROGRAMMATION SYSTOLIQUE
Christophe MAURAS
18 Pages, Juin 1989.

PI 483 POLYNOMIAL IDEAL THEORETIC METHODS IN DISCRETE EVENT, AND HYBRID DYNAMICAL SYSTEMS
Michel LE BORGNE, Albert BENVENISTE, Paul LE GUERNIC
20 Pages, Juillet 1989.

PI 484 IMPLEMENTING ATOMIC RENDEZVOUS WITHIN A TRANSACTIONAL FRAMEWORK
Jean-Pierre BANATRE, Michel BANATRE, Christine MORIN
22 Pages, Juillet 1989.

PI 485 THE MAPPING OF LINEAR RECURRENCE EQUATIONS ON REGULAR ARRAYS
Patrice QUINTON, Vincent VAN DONGEN
40 Pages, Juillet 1989.

Imprimé en France
par
l'Institut National de Recherche en Informatique et en Automatique